

# Apache Harmony

**Компоненты  
для  
переиспользования**

# Компоненты

- **Библиотека Java API (Classlibrary)**
- **Виртуальная машина (VM)**
  - Java-bytecode verifier (Verifier)
  - Обработчик системных исключений (Crash Handler)
  - Несколько различных компиляторов (JIT)
  - Расширяемый сборщик мусора (GC)
  - Интерпретатор (Interpreter)
  - Кодировщик Intel32 и Intel64 инструкций (Encoder)

# Успешное переиспользование

- **Classlibrary**

- ✓ Android, IBM JDK

- **Verifier**

- ✓ Eclipse

- **JIT**

- ✓ STM проект Intel
  - ✓ Moxie VM
  - ✓ Внутренние проекты Intel

- **Encoder**

- ✓ Различные студенческие проекты

# Classlibrary

- **Реализация Java API > 99%**
- **Независимость от VM**
  - DRLVM (Harmony)
  - IBM's VM
  - Google Android
- **Модульность**
  - Составные пакеты практически независимы и изначально были отдельными contributions

# Java Bytecode Verifier

- **Может быть переиспользован:**
  - Java bytecode анализ в не Java проектах
  - Для 100% поддержки всех данных class файлов
  - Переиспользован в *Eclipse* для отладки и инструментации Java кода
- **Характеристики**
  - Четко определенный интерфейс
    - В репозитории Harmony накопилось несколько альтернативных реализаций
  - Хорошая производительность
    - В разы быстрее SUN 1.6

# Crash Handler

- **Может быть переиспользован для обработки системных исключений:**
  - Поэтапный выход из программы
  - Автоматический вызов GDB
  - Печать стека и всех тредов
- **Характеристики**
  - Работает на Linux и Windows
  - Работает на 32 и 64 битных платформах
    - **Как и все остальное в Harmony!**

# Interpreter

- **Возможности переиспользования**
  - Прототипирование расширений Java на C++, без необходимости знания ассемблера:
    - Студенческие проекты
    - Поддержка новых языков программирования
    - Сбор различной профилировочной информации
- **Характеристики**
  - Малый размер исходного кода: ~200kb

# Garbage collector

- **Возможности переиспользования**

- Проверка новых методов “сборки мусора”
- Отличные от Java языки программирования
  - *Проект GSOC2008: Harmony GC будет интегрирован в ParrotVM (Tcl, Javascript, Ruby, Lua, Scheme, PHP, Python, Perl 6, APL, .NET)*

- **Характеристики**

- Спроектирован для удобства расширения:
  - ‘gc\_gen/src> ls’ :  
mark\_compact mark\_sweep move\_compact semi\_space  
trace\_forward ...



# JIT

- **Jitrino.JET – быстрый старт приложений**

- Минимум кода
- Быстрый запуск приложений:
  - Скорость компиляции: ~10 тыс. Java методов в секунду (1CPU 1.7Ghz)
- Скорость исполнения кода:
  - ~2-3 раза медленнее чем оптимизирующий компилятор
  - >10 раз быстрее известных нам интерпретаторов

- **Jitrino.OPT – максимум оптимизаций**

- Легкость прототипирования новых оптимизаций
  - Напишите один C++ класс реализующий оптимизацию
  - Добавьте имя оптимизации в конфигурацию запуска:  
`-XX:jit.CD_OPT.path.optimizer=ssa,devirt,hlo_api_magic,inline..`
- Расширяемость на другие языки/методы, опыт:
  - XSLT transformer
  - JavaScript
  - Transactional memory support

# С чего начать?

- **Изучите документацию на сайте**
  - <http://harmony.apache.org>
- **Подпишитесь на список рассылки**
  - Email: [dev@harmony.apache.org](mailto:dev@harmony.apache.org)
- **Спрашивайте любого из нас:**
  - Михаил Фурсов: [mike.fursov<at>gmail.com](mailto:mike.fursov@gmail.com)
  - Георгий Тимошенко: [george.timoshenko<at>gmail.com](mailto:george.timoshenko@gmail.com)
  - Алексей Варламов: [alexey.v.varlamov<at>gmail.com](mailto:alexey.v.varlamov@gmail.com)
  - Степан Мишура: [stepan.mishura<at>gmail.com](mailto:stepan.mishura@gmail.com)

# Вопросы

